

Security Scan Report

Scan Information

Target:	bennett.edu.in
Scan ID:	dff38e1e-f590-4aa8-b9ff-8b4b6c54a6bc
Date:	2025-04-28 04:03:58
Status:	N/A

Vulnerability Summary

Severity	Count
High	5
Medium	6
Low	1
Informational	3

Scan Results

Nmap Scan Results

Port	Protocol	State	Service	Product	Version
80	tcp	open	http-proxy	F5 BIG-IP load balancer http proxy	N/A
443	tcp	open	http	Apache httpd	N/A

Web Vulnerabilities

ID	Severity	Description
Target Host	Informational	bennett.edu.in
Target Port	Informational	80
GET /	Informational	The anti-clickjacking X-Frame-Options header is not present. See: https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Frame-Options :

GET /	Medium	The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fashion to the MIME type. See: https://www.netsparker.com/web-vulnerability-scanner/vulnerabilities/missing-content-type-header/ :
-------	--------	---

CVE Details

CVE ID	Severity	CVSS Score	Description
CVE-2023-29182	MEDIUM	5.5	A vulnerability in F5 BIG-IP load balancer http proxy allows attackers to perform privilege escalation via crafted requests.
CVE-1999-0236	HIGH	7.5	ScriptAlias directory in NCSA and Apache httpd allowed attackers to read CGI programs.
CVE-1999-0071	High	7.5	Apache httpd cookie buffer overflow for versions 1.1.1 and earlier.
CVE-2000-1206	Medium	5.0	Vulnerability in Apache httpd before 1.3.11, when configured for mass virtual hosting using mod_rewrite, or mod_vhost_alias in Apache 1.3.9, allows remote attackers to retrieve arbitrary files.
CVE-2002-1850	HIGH	7.5	mod_cgi in Apache 2.0.39 and 2.0.40 allows local users and possibly remote attackers to cause a denial of service (hang and memory consumption) by causing a CGI script to send a large amount of data to stderr, which results in a read/write deadlock between httpd and the CGI script.
CVE-2003-0249	High	7.5	PHP treats unknown methods such as "PoSt" as a GET request, which could allow attackers to intended access restrictions if PHP is running on a server that passes on all methods, such as Apache httpd 2.0, as demonstrated using a Limit directive. NOTE: this issue has been disputed by the Apache security team, saying "It is by design that PHP allows scripts to process any request method. A script which does not explicitly verify the request method will hence be processed as normal for arbitrary methods. It is therefore expected behaviour that one cannot implement per-method access control using the Apache configuration alone, which is the assumption made in this report.

CVE-2004-0493	Medium	6.4	The ap_get_mime_headers_core function in Apache httpd 2.0.49 allows remote attackers to cause a denial of service (memory exhaustion), and possibly an integer signedness error leading to a heap-based buffer overflow on 64 bit systems, via long header lines with large numbers of space or tab characters.
CVE-2004-2343	High	7.2	Apache HTTP Server 2.0.47 and earlier allows local users to bypass .htaccess file restrictions, as specified in httpd.conf with directives such as Deny From All, by using an ErrorDocument directive. NOTE: the vendor has disputed this issue, since the .htaccess mechanism is only intended to restrict external web access, and a local user already has the privileges to perform the same operations without using ErrorDocument
CVE-2005-3352	Medium	4.3	Cross-site scripting (XSS) vulnerability in the mod_imap module of Apache httpd before 1.3.35-dev and Apache httpd 2.0.x before 2.0.56-dev allows remote attackers to inject arbitrary web script or HTML via the Referer when using image maps.
CVE-2005-3630	Medium	5.0	Fedora Directory Server before 10 allows remote attackers to obtain sensitive information, such as the password from adm.conf via an IFRAME element, probably involving an Apache httpd.conf configuration that orders "allow" directives before "deny" directives.
CVE-2006-4625	Low	3.6	PHP 4.x up to 4.4.4 and PHP 5 up to 5.1.6 allows local users to bypass certain Apache HTTP Server httpd.conf options, such as safe_mode and open_basedir, via the ini_restore function, which resets the values to their php.ini (Master Value) defaults.

Recommendations

- Keep all software up-to-date with the latest security patches
- Implement proper network segmentation to limit access to sensitive systems
- Use strong password policies and consider multi-factor authentication
- Monitor system logs for suspicious activities
- Regularly perform security scans and penetration testing
- Follow the principle of least privilege for user access